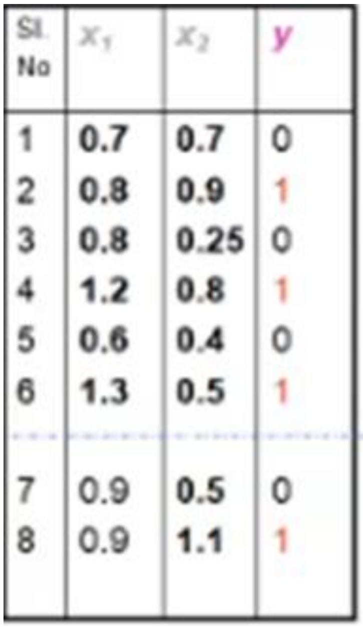
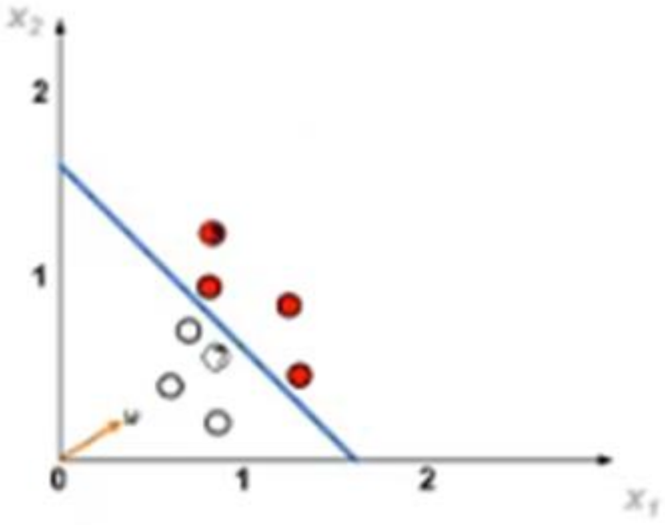
tutorial

1. Consider a problem with two features (x_1 and x_2).



Sl. No	x_1	x_2	y
1	0.7	0.7	0
2	0.8	0.9	1
3	0.8	0.25	0
4	1.2	0.8	1
5	0.6	0.4	0
6	1.3	0.5	1
7	0.9	0.5	0
8	0.9	1.1	1



The equation of the decision boundary is

$y = x_1 + x_2 - 1.5$ or $y = wx + b$. This is called 'linear discriminant function' in statistical studies.

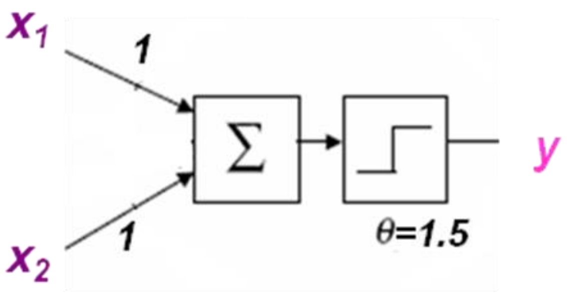
Or the output $y = f(x, w, b)$, where x is the input, w and b are parameters.

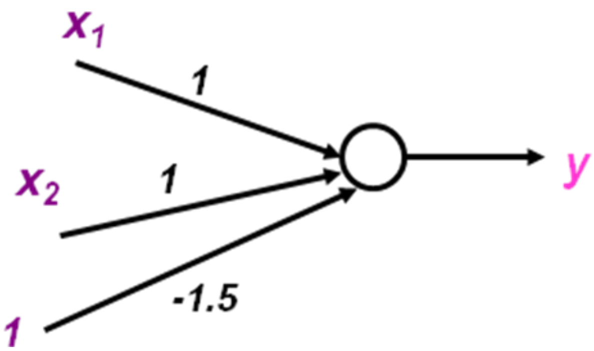
w is called weights in neural network terminology and *b* is called the bias. –*b* is called threshold.

Note that the weight vector decides the orientation of the decision surface and bias decides the distance from the origin.

In ***classification*** problem output *y* is one of a number of discrete classes or categories. But in a **regression** problem, output *y* represents the values of continuous variables.

Then we studied, how we can solve the problem in a 'neural network' way



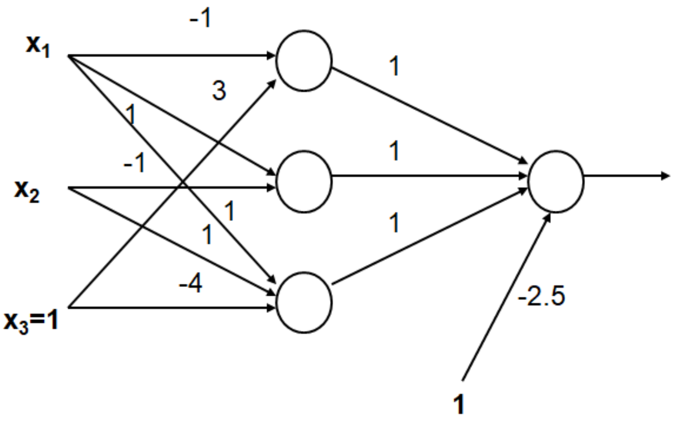


The second figure is a neural work with a single neuron. What does the neuron do? It sums up all the inputs then produces an output.

Now copy the following lines into a matlab file and run (or create code using python). This is your first neural network!

```
x=input('x=');
sum=x(1)+x(2);
if sum>=1.5
    out=1;
else out=0;
end
if out==1
disp('diseased')
else
disp('no disease')
end
Then write in the command window x=[.6 .7]
```

2. Find the output of the following perceptron neural network for the inputs [1 1] and [1 0]



3. Gaussian membership function is defined by *c* and *σ*. $y = e^{-0.5\left(\frac{x-c}{\sigma}\right)^2}$

   Create a Gaussian membership function with centre as 40 and spread 10 (x varies from 0 to 100)

   a. Plot the above function.

   b. Create a general function for the Gaussian function.

4. Study the command ***newp*** **in** matlab and solve examples using this command (or study a possible toolbox in python, use keras)**.** You can solve OR gate.

5. **Visualization of data:** Load fisheriris data and plot the data in 2-D and 3-D

```
load fisheriris
gscatter(meas(:,1), meas(:,2), species,'rgb','osd');
xlabel('Sepal length');
ylabel('Sepal width');
```

6. Write a MATLAB/Python function for perceptron learning rule (PLR)

7. Write a MATLAB/Python function for gradient descent learning rule